



Feedback



Report a problem



Satisfaction

survey

THE ACM DIGITAL LIBRARY



[Feedback](#) [Report a problem](#) [Satisfaction](#)

Terms used: [Binary matching tool](#)

Found 36,559 of 216,199

Sort results by [relevance](#)

[Save results to a Binder](#)

[Try an Advanced Search](#)

Display results [expanded form](#)

[Search Tips](#)

[Try this search in The ACM Guide](#)

[Open results in a new window](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale

- 1 [Fine grained data management to achieve evolution resilience in a software development environment](#)

[development environment](#)

Richard Snodgrass, Karen Shannon

October 1990 **ACM SIGSOFT Software Engineering Notes , Proceedings of the fourth ACM SIGSOFT symposium on Software development environments SDE 4**, Volume 15 Issue 6

Publisher: ACM Press

Full text available: [pdf\(1.72 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

A software development environment (SDE) exhibits evolution resilience if changes to the SDE do not adversely affect its functionality nor performance, and also do not introduce delays in returning the SDE to an operational state after a change. Evolution resilience is especially difficult to achieve when manipulating fine grained data, which must be tightly bound to the language in which the SDE is implemented to achieve adequate performance. We examine a spectrum of approaches to tool int ...

- 2 [Effectively prioritizing tests in development environment](#)

[Amitabh Srivastava, Jay Thiagarajan](#)

July 2002 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis ISSTA 02**, Volume 27 Issue 4

Publisher: ACM Press

Full text available: [pdf\(381.17 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Software testing helps ensure not only that the software under development has been implemented correctly, but also that further development does not break it. If developers introduce new defects into the software, these should be detected as early and inexpensively as possible in the development cycle. To help optimize which tests are run at what points in the design cycle, we have built

repositories MSR '06**Publisher:** ACM PressFull text available:  pdf(154.94 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Multi-version program analyses require that elements of one version of a program be mapped to the elements of other versions of that program. Matching program elements between two versions of a program is a fundamental building block for multi-version program analyses and other software evolution research such as profile propagation, regression testing, and software version merging. In this paper, we survey matching techniques that can be used for multi-version program analyses and evaluate them ...

Keywords: matching, multi-version analysis, software evolution

- 4 **Reuse and OTS: Pallino: automation to support regression test selection for cots-based applications** 

Jiang Zheng, Laurie Williams, Brian Robinson

November 2007 **Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering ASE '07****Publisher:** ACMFull text available:  pdf(233.61 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Software products are often built from commercial-off-the-shelf (COTS) components. When new releases of these components are made available for integration and testing, source code is usually not provided by the vendors. Various regression test selection techniques have been developed and have been shown to be cost effective. However, the majority of these test selection techniques rely on source code for change identification and impact analysis. In our research, we have evolved a regression ...

Keywords: COTS, commercial-off-the-shelf, regression testing, software testing

- 5 **WBIA'05: Practical analysis of stripped binary code** 

Laune C. Harris, Barton P. Miller

December 2005 **ACM SIGARCH Computer Architecture News**, Volume 33 Issue 5**Publisher:** ACM PressFull text available:  pdf(224.79 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Executable binary code is the authoritative source of information about program content and behavior. The compile, link, and optimize steps can cause a program's detailed execution behavior to differ substantially from its source code. Binary code analysis is used to provide information about a program's content and structure, and is therefore a foundation of many applications, including binary modification[3,12,22,31], binary translation[5,29], binary matching[30], performance profiling[13,16,1 ...]

- 6 **1994: ATOM: a system for building customized program analysis tools** 

Amitabh Srivastava, Alan Eustace

April 2004 **ACM SIGPLAN Notices**, Volume 39 Issue 4**Publisher:** ACM PressFull text available:  pdf(1.72 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

ATOM (**A**nalysis **T**ools with **O**M) is a single framework for building a wide range of customized program analysis tools. It provides the common infrastructure present in all code-instrumenting tools; this is the difficult and time-consuming part. The user simply defines the tool-specific details in instrumentation and analysis routines. Building a basic block counting tool like Pixie with ATOM requires only a page of code. ATOM, using OM link-time technology, organizes the ...

7 Matching execution histories of program versions Xiangyu Zhang, Rajiv GuptaSeptember 2005 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering ESEC/FSE-13, Volume 30 Issue 5****Publisher:** ACM PressFull text available:  pdf(247.29 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We develop a method for matching dynamic histories of program executions of two program versions. The matches produced can be useful in many applications including software piracy detection and several debugging scenarios. Unlike some static approaches for matching program versions, our approach does not require access to source code of the two program versions because dynamic histories can be collected by running instrumented versions of program binaries. We base our matching algorithm on compa ...

Keywords: debugging, dynamic analysis, execution traces, piracy detection**8 Experience papers: software development practices: Applying regression test** selection for COTS-based applications

Jiang Zheng, Brian Robinson, Laurie Williams, Karen Smiley

May 2006 **Proceeding of the 28th international conference on Software engineering ICSE '06****Publisher:** ACM PressFull text available:  pdf(275.40 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

ABB incorporates a variety of commercial-off-the-shelf (COTS) components in its products. When new releases of these components are made available for integration and testing, source code is often not provided. Various regression test selection processes have been developed and have been shown to be cost effectiveness. However, the majority of these test selection techniques rely on access to source code for change identification. In this paper we present the application of the lightweight Integ ...

Keywords: COTS, commercial-off-the-shelf, regression testing, software testing**9 Industrial, application, and experience sessions: profiling: Bridging the application and DBMS profiling divide for database application developers**

Surajit Chaudhuri, Vivek Narasayya, Manoj Syamala

September 2007 **Proceedings of the 33rd international conference on Very large data bases VLDB '07****Publisher:** VLDB EndowmentFull text available:  pdf(578.67 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

In today's world, tools for profiling and tuning application code remain disconnected from the profiling and tuning tools for relational DBMSs. This makes it challenging for developers of database applications to profile, tune and debug their applications, for example, identifying application code that causes deadlocks in the server. We have developed an infrastructure that simultaneously captures *both the application context as well as the database context*, thereby enabling a rich cla ...

10 Overcoming the challenges to feedback-directed optimization (Keynote Talk)

Michael D. Smith

January 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN workshop on**



Dynamic and adaptive compilation and optimization DYNAMO '00, Volume 35 Issue 7

Publisher: ACM Press

Full text available: pdf(1.33 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Feedback-directed optimization (FDO) is a general term used to describe any technique that alters a program's execution based on tendencies observed in its present or past runs. This paper reviews the current state of affairs in FDO and discusses the challenges inhibiting further acceptance of these techniques. It also argues that current trends in hardware and software technology have resulted in an execution environment where immutable executables and traditional static optimizations are ...

11 Profile-based optimizations: Reality-based optimization



Scott McFarling

March 2003 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization CGO '03**

Publisher: IEEE Computer Society

Full text available: pdf(1.09 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Profile-based optimization has been studied extensively. Numerous papers and real systems have shown substantial improvements. However, most of these papers have been limited to either branch prediction or instruction cache performance. Also, most of these papers have looked at small applications with a limited number of testing and training scenarios. In this paper, we look at real use of large real-world desktop applications. We also assume memory consumption and disk performance are the primar ...

12 Fast address lookups using controlled prefix expansion



V. Srinivasan, G. Varghese

February 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 1

Publisher: ACM Press

Full text available: pdf(258.60 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Internet (IP) address lookup is a major bottleneck in high-performance routers. IP address lookup is challenging because it requires a longest matching prefix lookup. It is compounded by increasing routing table sizes, increased traffic, higher-speed links, and the migration to 128-bit IPv6 addresses. We describe how IP lookups and updates can be made faster using a set of transformation techniques. Our main technique, controlled prefix expansion, transf ...

Keywords: Internet address lookup, binary search on levels, controlled prefix expansion, expanded tries, longest-prefix match, multibit tries, router preformance

13 The emporium approach to computer science education



J. A. N. Lee

June 2002 **ACM SIGCSE Bulletin , Proceedings of the 7th annual conference on Innovation and technology in computer science education ITiCSE '02**,

Volume 34 Issue 3

Publisher: ACM Press

Full text available: pdf(220.44 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

As an alternative to laboratories for computer science programming courses, the Emporium style of learning environment has advantages for a course concerned with the foundations and principles of computer science. Following the lead from courses in freshman mathematics, this paper describes the active learning and assessment

methodologies incorporated into a freshman computer science course. The results have been encouraging, and student acceptance of an alternative approach to learning is impro ...

Keywords: curriculum, education, emporium, laboratories, on-line

14 [Ontologies: Constructing virtual documents for ontology matching](#)

 Yuzhong Qu, Wei Hu, Gong Cheng

May 2006 **Proceedings of the 15th international conference on World Wide Web
WWW '06**

Publisher: ACM Press

Full text available: .pdf(283.69 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

On the investigation of linguistic techniques used in ontology matching, we propose a new idea of virtual documents to pursue a cost-effective approach to linguistic matching in this paper. Basically, as a collection of weighted words, the virtual document of a URIref declared in an ontology contains not only the local descriptions but also the neighboring information to reflect the intended meaning of the URIref. Document similarity can be computed by traditional vector space techniques, and th ...

Keywords: description, formulation, linguistic matching, neighboring operation, ontology matching, vector space model

15 [Tree-based mapping of algorithms to predefined structures](#)

Peter Marwedel

November 1993 **Proceedings of the 1993 IEEE/ACM international conference on
Computer-aided design ICCAD '93**

Publisher: IEEE Computer Society Press

Full text available: .pdf(693.29 KB) Additional Information: [full citation](#), [references](#), [citations](#)

16 [Ubicomp tools: Authoring sensor-based interactions by demonstration with direct
manipulation and pattern recognition](#)

 Björn Hartmann, Leith Abdulla, Manas Mittal, Scott R. Klemmer

April 2007 **Proceedings of the SIGCHI conference on Human factors in computing
systems CHI '07**

Publisher: ACM Press

Full text available: .pdf(1.18 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Sensors are becoming increasingly important in interaction design. Authoring a sensor-based interaction comprises three steps: choosing and connecting the appropriate hardware, creating application logic, and specifying the relationship between sensor values and application logic. Recent research has successfully addressed the first two issues. However, linking sensor input data to application logic remains an exercise in patience and trial-and-error testing for most designers. This paper int ...

Keywords: design tools, pbd, physical computing, sensors

17 [Industrial sessions: beyond relational tables: An ebXML infrastructure implementation](#)

 through UDDI registries and RosettaNet PIPs

Asuman Dogac, Yusuf Tambag, Pinar Pembecioglu, Sait Pektas, Gokce Laleci, Gokhan Kurt, Serkan Toprak, Yildiray Kabak

June 2002 **Proceedings of the 2002 ACM SIGMOD international conference on**

Management of data SIGMOD '02**Publisher:** ACM PressFull text available:  pdf(1.25 MB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Today's Internet based businesses need a level of interoperability which will allow trading partners to seamlessly and dynamically come together and do business without ad hoc and proprietary integrations. Such a level of interoperability involves being able to find potential business partners, discovering their services and business processes, and conducting business "on the fly". This process of dynamic interoperation is only possible through standard B2B frameworks. Indeed a number of B2B ele ...

18 Binary synthesis 

Greg Stitt, Frank Vahid

August 2007 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 12 Issue 3**Publisher:** ACM PressFull text available:  pdf(341.48 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Recent high-level synthesis approaches and C-based hardware description languages attempt to improve the hardware design process by allowing developers to capture desired hardware functionality in a well-known high-level source language. However, these approaches have yet to achieve wide commercial success due in part to the difficulty of incorporating such approaches into software tool flows. The requirement of using a specific language, compiler, or development environment may cause many so ...

Keywords: Binary synthesis, FPGA, configurable logic, hardware/software codesign, hardware/software partitioning, synthesis from software binaries, warp processors

19 An innovative simulation tool for advanced signal processing in UMTS systems 

Dania Marabissi, Marco Michelini, Luca Simone Ronga

September 2004 **Wireless Networks**, Volume 10 Issue 5**Publisher:** Kluwer Academic PublishersFull text available:  pdf(545.12 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Link-level simulations are essential in the design of UMTS communication systems. The large number of interdependent variables makes it impossible to derive easy design steps without an efficient modeling of the environments and the implemented reception schemes. In this paper, a novel tool for UMTS design is presented. The tool includes a fast C++ simulation engine and a complete 3GPP library to model the uplink transmission chain. As an example, a series of Monte Carlo performance simulatio ...

Keywords: 3G-simulation environment, CDMA advanced receivers, DSP system design, code division multiple access (CDMA), fading channel models, multirate systems, object-oriented simulation tool

20 A guided tour to approximate string matching 

Gonzalo Navarro

March 2001 **ACM Computing Surveys (CSUR)**, Volume 33 Issue 1**Publisher:** ACM PressFull text available:  pdf(1.19 MB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

We survey the current techniques to cope with the problem of string matching that allows errors. This is becoming a more and more relevant issue for many fast growing areas such

as information retrieval and computational biology. We focus on online searching and mostly on edit distance, explaining the problem and its relevance, its statistical behavior, its history and current developments, and the central ideas of the algorithms and their complexities. We present a number of experiments to ...

Keywords: Levenshtein distance, edit distance, online string matching, text searching allowing errors

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)